

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Суннатилло Махмудов¹

22 апреля, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Изучить основы программирования в оболочке ОС UNIX.
Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Задачи лабораторной работы

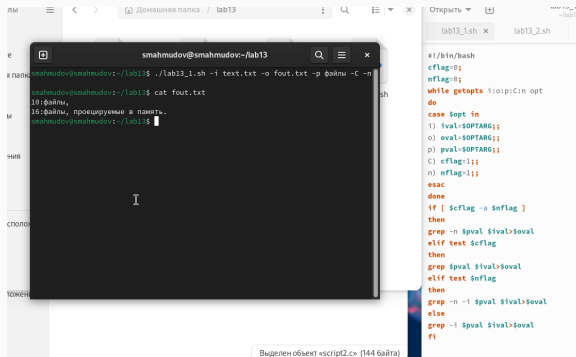
1 Выполнить 4 задания

Процесс выполнения лабораторной работы

1. Используя команды `getopts` `grep` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

Выполнение работы



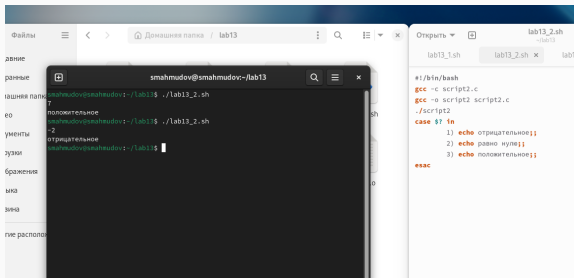
The image shows a screenshot of a computer screen with a terminal window and a file editor. The terminal window, titled 'smahmudov@smahmudov:~/lab13', shows the execution of a script 'lab13_1.sh' with arguments '-i text.txt -o fout.txt -p файлы -C -n'. The output of the script is displayed in the terminal: '18:файлы,' and '16:файлы, проецируемые в память.' The file editor, titled 'lab13_1.sh', shows the source code of the script. The script is a shell script that takes several arguments and processes them. It uses a 'while' loop to iterate over the arguments and a 'case' statement to handle different options. The script also uses 'grep' to search for patterns in the input files.

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts f:io:p:C:n opt
do
case $opt in
f) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
fi
```

Рис. 1: Задание 1

2. Напишем сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено

Выполнение работы



```
smahmudov@smahmudov:~/lab13$ ./lab13_2.sh
положительное
smahmudov@smahmudov:~/lab13$ ./lab13_2.sh
-2
smahmudov@smahmudov:~/lab13$ ./lab13_2.sh
отрицательное
```

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```

Рис. 2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

Выполнение работы

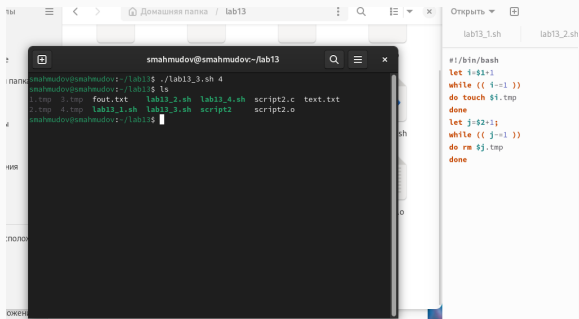
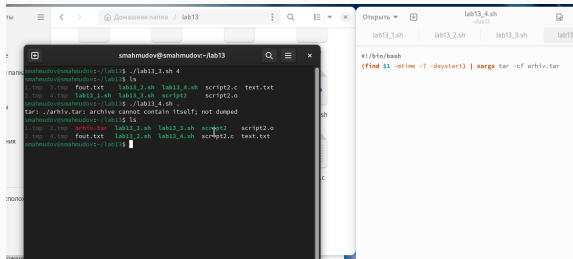


Рис. 3: Задание 3

4. Напишем командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

Выполнение работы



The image shows a terminal window and a file explorer. The terminal window, titled 'smahmudov@smahmudov:~/lab13', displays the following commands and output:

```
smahmudov@smahmudov:~/lab13 ./lab13_3.sh 4
smahmudov@smahmudov:~/lab13 ls
1. tmp 3. tmp fout.txt lab13_3.sh lab13_4.sh script2.c text.txt
2. tmp 4. tmp lab13_1.sh lab13_2.sh script2 script2.o
smahmudov@smahmudov:~/lab13 ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
smahmudov@smahmudov:~/lab13 ls
1. tmp 3. tmp arhiv.tar lab13_3.sh lab13_4.sh script2 script2.o
2. tmp 4. tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
smahmudov@smahmudov:~/lab13
```

The file explorer, titled 'lab13_4.sh', shows the following content:

```
#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

Рис. 4: Задание 4

Выводы по проделанной работе

В данной работе мы изучили основы программирования в оболочке ОС UNIX и писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.